IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): Christopher William Preist, et al.	Group: unknown
	Examiner: unknown
Serial No.: 10/717,206	
	Our Ref: B-5307 621517-4
Filed: November 18, 2003	
For: "A METHOD OF SYSTEM FOR INTEGRATING INTERACTION PROTOCOLS BETWEEN TWO ENTITIES"	Date: March 25, 2004

CLAIM TO PRIORITY UNDER 35 U.S.C. 119

MAIL STOP MISSING PARTS
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

[X] Applicants hereby make a right of priority claim under 35 U.S.C. 119 for the benefit of the filing date(s) of the following corresponding foreign application(s):

COUNTRY	FILING DATE	SERIAL NUMBER
GB	18 November 2002	0226778.9

[ ] A certified copy of each of the above-noted patent applications was filed with the Parent Application No.________________________.

[X] To support applicant's claim, a certified copy of the above-identified foreign patent application is enclosed herewith.

[ ] The priority document will be forwarded to the Patent Office when required or prior to issuance.

I hereby certify that this correspondence is being deposited with the United States Postal Service with sufficient postage as first-class mail in an envelope addressed to the "Mail Stop Missing Parts, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450", on March 25, 2004

Respectfully submitted,

USSN 10/717,206

The Patent Office

INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely subjects the company to certain additional company law rules.

Signed

Dated 4 December 2003

Patents Form 1/77

Patents Act 1977
(Rule 16)

18NOV02 E764130-1 D01463
P01/7700 0.00-0226778.9

THE PATENT OFFICE
A
18 NOV 2002
RECEIVED BY FAX

The Patent Office

Cardiff Road
Newport
South Wales
NP10 8QQ

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

1. Your reference — 200300288-01 GB

18 NOV 2002

2. Patent application number *(The Patent Office will fill in this part)* — 0226778.9

3. Full name, address and postcode of the or of each applicant *(underline all surnames)*

Hewlett-Packard Company
3000 Hanover Street
Palo Alto
CA 94304, USA

Patents ADP number *(if you know it)* — 4965588001

If the applicant is a corporate body, give the country/state of its incorporation — Delaware, USA

4. Title of the invention — Using Semantic Web Technology To Enhance Current Business-to-Business Integration Approaches

5. Name of your agent *(if you have one)*

"Address for service" in the United Kingdom to which all correspondence should be sent *(including the postcode)*

Chris Harrison
Hewlett-Packard Ltd, IP Section
Filton Road, Stoke Gifford
Bristol BS34 8QZ

Patents ADP number *(if you know it)* — 8191439001

6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and *(if you know it)* the or each application number

Country	Priority application number *(if you know it)*	Date of filing *(day / month / year)*

7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application

Number of earlier application	Date of filing *(day / month / year)*

8. Is a statement of inventorship and of right to grant of a patent required in support of this request? *(Answer 'Yes' if:*
a) any applicant named in part 3 is not an inventor, or
b) there is an inventor who is not named as an applicant, or
c) any named applicant is a corporate body.
See note (d))

Yes

Patents Form 1/77

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form	
Description	7
Claim(s)	
Abstract	
Drawing(s)	

10. If you are also filing any of the following, state how many against each item.

Priority documents	
Translations of priority documents	
Statement of inventorship and right to grant of a patent *(Patents Form 7/77)*	
Request for preliminary examination and search *(Patents Form 9/77)*	
Request for substantive examination *(Patents Form 10/77)*	
Any other documents *(please specify)*	

11. I/We request the grant of a patent on the basis of this application.

Signature Date 19 /11/2002

12. Name and daytime telephone number of person to contact in the United Kingdom — Tony Judd Tel: 0117-312-8026

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

a) *If you need help to fill in this form or you have any questions, please contact the Patent Office on 08459 500505.*

b) *Write your answers in capital letters using black ink or you may type them.*

c) *If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*

d) *If you have answered 'Yes' Patents Form 7/77 will need to be filed.*

e) *Once you have filled in the form you must remember to sign and date it.*

f) *For details of the fee and ways to pay please contact the Patent Office.*

Using Semantic Web Technology to Enhance Current Business-to-Business Integration Approaches

David Trastour
HP labs
Filton Rd, Stoke Gifford
Bristol, BS32 8QZ
United Kingdom
david.trastour@hp.com

Chris Preist
HP labs
Filton Rd, Stoke Gifford
Bristol, BS32 8QZ
United Kingdom
chris.preist@hp.com

Derek Coleman
RosettaNet/HP
19111 Pruneridge Ave.
Cupertino, CA 95014
United States
derek.coleman@hp.com

ABSTRACT

Setting up electronic Business-to-Business relationships is time-consuming and costly. It has been eased to a certain extent by standards such as RosettaNet, which use XML and XML Schema technologies to define standardised syntax of messages used in interactions. However, this standardisation has necessarily maintained some flexibility to allow companies with different internal processes to comply with the standard. Furthermore, the standard is syntactic, rather than semantic. Semantic constraints on interactions are currently represented informally.

In this paper, we describe an application of Semantic Web technology to enhance RosettaNet and further reduce cost and time. Businesses can represent the possible ways they are able to interact as semantic and syntactic constraints. Two businesses can determine if they are able to interact without altering their business process by sharing constraints, and finding if the overall set is satisfiable. If it is not, they can use the data to determine what changes need to be made to their business processes. They can also use the other business' constraints to verify or generate documents which meet the constraints, and so are usable by the other business. The system integrates with current RosettaNet standards and tools through the use of a translation suite able to transform XML Schema into DAML+OIL and XML into RDF.

Categories and Subject Descriptors

E.1 [**Data**]: Data Structures; I.2.4 [**Computing Methodologies**]: Artificial Intelligence—*Knowledge Representation Formalisms and Methods*; J.1 [**Computer Applications**]: Administrative Data Processing

General Terms

Languages, Standardization

Keywords

Semantic Web, Business-to-Business, Interoperability, DAML+OIL, XML Schema

1. INTRODUCTION

Despite the bursting of the dot com bubble, electronic commerce continues to be an increasingly important aspect of the economy. To the general public, the most visible face is the increasing number of Business-to-Consumer interactions available via the web. However, the majority of economic transactions occur between businesses, making the Business-to-Business (B2B) aspect of electronic commerce a significantly larger market. Historically, business relationships have been long-term. When such a relationship has been established, EDI (Electronic Data Interchange) technology can be used to automate certain straightforward interactions between the business partners – for example, the purchase of goods at a pre-agreed price, and the delivery of them. Setting up the EDI system requires the two parties to agree on interaction protocols and message formats, and to implement a messaging system that meets these agreements – often over a Virtual Private Network. This can be a time-consuming and costly process.

RosettaNet [15] is an industrial consortium which aims to make this process cheaper and more straightforward, by using XML [3] messaging technology transported over the World Wide Web. It does this by standardising the format, content and sequence of messages between partners for a variety of possible interactions which companies can use in B2B relationships. Hence, companies do not need to go through a lengthy negotiation to specify the way in which they are going to interact. Instead, they simply need to agree on which standard interaction to use. Standardisation also speeds up the development process, as products such as WebMethods Trading Networks Server include software libraries and XML templates supporting RosettaNet interactions.

This standardisation effort has substantially reduced the cost and time of setting up a B2B relationship. However, because it is based on XML technology, the tools provided are primarily syntactic. In this paper, we describe an application of Semantic Web technology to enhance RosettaNet and further reduce cost and time. Businesses can represent the possible ways they are able to interact as semantic constraints. Two businesses can determine if they are able to interact without altering their business process by sharing constraints, and finding if the overall set is satisfiable. They can use the other business' constraints to generate documents which meet the constraints, and so are useable by the other business.

This application is evolutionary, rather than revolutionary. It accepts existing RosettaNet design decisions and tools as they currently are, rather than requiring modification or re-design of these. As a result of this, it is more likely to be rapidly accepted and adopted by the current B2B developer community. This means it must use Semantic Web technology in a relatively conservative way. We believe that by doing this the developer community will become more familiar with the basic ideas, allowing more radical approaches to be taken in the future.

The structure of this paper is as follows. In section 2 we give

WWW2003, May 20–24, 2003, Budapest, Hungary.
ACM xxx.

an overview of RosettaNet and present some of the problems that implementors encounter when they deploy RosettaNet solutions. In section 3 we describe the Nile system which aims at solving these problems with Semantic Web techniques. In particular we describe the two main components of this system, the Nile translation suite and the Constraint Knowledge Base. In section 4 we describe a typical use of the Nile system and show how it solves the problems exposed in section 2. We then discuss related work (section 5) and we conclude presenting our future work intentions (section 6).

2. ROSETTANET NOW

RosettaNet standards have gone a long way to ease the process of setting up and executing long-term B2B relationships via the web. The key concept used to do this is the Partner Interface Process (PIP). PIPs are used to define standard ways of interacting between companies to carry out a specified task. They define the aspects of a business process which are common to the two parties, but place no constraints on how the internal processes implement these common aspects. A PIP specification defines the flow of message documents which will take place during an interaction, and also specifies the format of the messages. A message format is defined through 'message guidelines' documentation, and an XML DTD describing the syntactic structure a message should have.

Hence, in theory, all businesses have to do to set up a new partnership is to agree on which PIPs to use, and implement the PIPs according to their specification. However, as different businesses can have different back-end processes, some flexibility within the standards is necessary to enable all businesses to satisfy it. For example, one business may normally represent dates on invoices using ISO 8601 format (YYYY-MM-DD) while another may use UK common practice (DD/MM/YYYY). One business may expect the account details of the buyer on an invoice, while another may not. To allow differences such as these, PIP definitions often make use of generic datatypes (such as strings or integers) and include optional fields or fields with unbounded cardinalities. As a result of this flexibility, there is no guarantee that two RosettaNet compliant companies will be able to communicate with each other: different business practices or back-end systems may impose different conditions on the presence of some information or on its format. Because of this, it is necessary to reconcile the different processes used by two companies which intend to interact via RosettaNet. There is some flexibility in the way in which a PIP can be implemented, and it is necessary that interacting parties agree as to the specific implementation chosen. This process of reconciliation is currently carried out off-line, using spreadsheets to document decisions. Developers then implement these decisions as they encode the PIPs. This can be a very time-consuming process, meaning that it can take many months to create a new RosettaNet partnership. Hence interoperability, one of the advantages of standardisation, is sacrificed in favour of flexibility.

RosettaNet are currently developing Next Generation PIPs [] in an attempt to produce specifications than are more formal than the message guidelines used in the current standards. For each Next Generation PIP, RosettaNet specifies a UML class diagram [13] and XML Schemas [17, 2] that replace the XML DTDs. The UML class diagram defines the business objects – such as financial documents or purchase requests – that are used in the PIP. To encourage reuse across PIPs, RosettaNet defines a domain model, i.e. a set of base classes that can be reused or subclassed in the UML class diagrams. The XML Schemas define what makes an XML document a syntactically valid PIP document. XML Schemas are currently defined manually from the UML class diagram.

Having an explicit machine-readable representation of the constraints imposed by a PIP makes setting up a partnership quicker and easier. Reconciliation can take place by agreeing a set of further constraints on each XML Schema within the PIP. Furthermore, having the agreed document structure specified in this format allows the developers to use tools such as Contivo [5] to rapidly automate the process of document generation.

However, this approach has several disadvantages:

- The constraints that XML Schema are able to represent are mainly constraints on the syntax, not the semantics, of documents. This means certain constraints which appear in a PIP specification which cannot be represented in the XML Schema. A typical example of this sort of constraint is a dependency between fields, for instance the presence of a field implying a cardinality constraint on another field. As an experiment, RosettaNet is using OCL [12] to represent such constraints in the definition of Next Generation PIPs. Currently, these are documented as comments within the XML Schemas.
- As seen earlier, a company's business processes impose constraints on the deployment of PIPs. Some of these constraints are of syntactic nature and can usually be captured in an XML Schema – which must be more specific than the PIP XML Schema. Some may be of semantic nature and so cannot be expressed in XML Schema. Companies deploying RosettaNet PIPs usually document these constraints in the form of spreadsheets that are manually created for the purpose of one deployment.
- Additional constraints imposed during the reconciliation process may also be semantic in nature, and therefore cannot currently be represented in a machine-readable format.
- The same business object class may appear in several documents exchanged during an interaction. Constraints imposed on this class should be applied to all documents that use this class (either directly or through a subclass). Currently, this will mean editing all of the associated schema to include the constraint. This imposes an unnecessary burden on the developers, and can potentially pose maintenance problems.
- Constraints on a business object class depend on the context – i.e. the specific deployment scenario – this class is used. We have identified that the deployment context is a function of:
  1. the PIP Document being used;
  2. the trading partner one is doing business with and whether they act as a buyer or seller;
  3. the business process being used(different business processes using a given PIP with a given trading partner may impose different constraints because of business requirements of back-end systems).

The Next Generation PIP cannot adequately manage the application of different constraints in different circumstances. Currently, the developers would have to manually aggregate the constraints corresponding to a deployment context into refined XML Schemas and other informal documents – when XML Schema is not expressive enough. This is inefficient and could pose maintenance problems. Moreover, since these constraints are not captured in a formal and systematic way, some knowledge could be lost from a deployment to the next.

- The same constraint may apply to a certain class of partners. For instance, a back-end system could impose a constraint on a *Tax* class for all its European partners. Similarly, it should be possible to apply constraints on classes of PIP documents (e.g. Invoicing documents) or business processes (e.g. Electronic component purchasing).

In this paper, we present a way of overcoming these shortfalls. We present a system that can manage the relationship between a business and several different partners by formally capturing the constraints on RosettaNet deployments. It is able to automatically detect if interactions are possible with a new potential partner, and can support the reconciliation process by allowing implementation of both syntactic and semantic constraints agreed by the partners. The system is able to determine exactly which constraints to apply depending on the deployment context, and is able to automatically generate an appropriate schema to communicate with the business partner.

3. THE NILE SYSTEM: INTRODUCING SEMANTICS

We have developed the Nile system to ease the B2B integration process, and to overcome the shortfalls of RosettaNet outlined above. In this system, we rely on XML Schema to express the syntactic constraints and on DAML+OIL [9] to express the semantic constraints. Our approach makes use of XML Schema to define and validate the syntax of PIP documents (in particular XML Schema can check the ordering of fields withing a document and the format of datatypes [2]). It also makes use of DAML+OIL to define semantic constraints. As we will show in the remainder of this paper, we believe that DAML+OIL can be used to model:

1. the business object class hierarchies and their attributes (or properties in Semantic Web terms);
2. the semantic constraints on business objects from the PIP definitions (currently modelled in OCL);
3. the notion of deployment context;
4. the additional semantic constraints imposed by a business with respect to a deployment context.

To our knowledge current B2B standards do not make use of DAML+OIL. However they often rely on XML Schema to define the syntax of the documents being exchanged. There is hence a need to lift the B2B schemas to an ontology language.

Nile consists of three key technology components:

- The XML Schema to DAML+OIL translation tool which converts XML Schemas into DAML+OIL class hierarchies and constraints;
- The Constraint Knowledge Base. This is a structured knowledge base, in DAML+OIL, which describes the constraints that a business places on business object classes and in what deployment context these constraints apply. The Nile Constraint Editor manipulates the Constraint Knowledge Base and it allows a user:
    1. to populate the deployment context ontology, i.e. to define the instances and classes representing the set of PIP documents, partners and business processes which characterise a business' RosettaNet deployments;
    2. to browse the business object class definitions in the DAML+OIL knowledge base;
    3. to create, modify and browse constraints on business objects in a given deployment context.

  The Nile Constraint Editor provides a set of constraint templates of the forms typically encountered in RosettaNet implementations. A business will create constrains using these templates.
- The XML document Validator. This set of generic tools is used for translating documents from the 'syntactic' world of XML into the 'semantic' world of RDF [10]. Specifically, it's functionality allows:
    1. A 'best effort' translation of DAML+OIL class hierarchies and constraints back into XML Schema;
    2. XML documents to be translated into RDF.

3.1 XML Schema to DAML+OIL mapping tool

Many XML based applications, like RosettaNet, would benefit from rich semantic modelling capabilities. While XML Schema covers some simple data modeling needs [11], we would like to get access to more general and powerful techniques.

XML defines a transfer syntax for tree-structured documents. XML Schema definitions holds the declarations for validating XML instance documents. These declarations are syntactic constraints on what make a valid XML document.

In the Semantic Web domain, RDF [10] models data in the form of directed labeled graphs and is layered on top of XML for serialization. As pointed out in [14], this choice of a different data model makes rich semantic descriptions and inferencing are out of reach for XML applications. In particular, we would like to benefit from DAML+OIL [9] modeling capabilities in B2B applications such as RosettaNet since it offers an expressive logic while keeping efficient reasoning possible [7].

The XML Schema to DAML+OIL mapping tool generates a DAML+OIL ontology from an XML Schema type hierarchy. The purpose of such a tool is to lift XML Schema to the level of an ontology. It creates a skeleton ontology which can be extended with any DAML+OIL editor (such as OilEd [1]). In the context of Nile, we use this tool to generate a base ontology from the XML Schema provided by RosettaNet: each business object class has a syntactic definition (its XML Schema type) and a semantic definition (its associated DAML+OIL class). We can then augment this ontology with additional constraints.

We present an overview of the mapping from XML Schema to DAML+OIL in figure 1. An exhaustive discussion on this mapping is out scope of this paper.

The following is an example of taken from PIP3C3 [16] which is a notification of Invoice process. From the $PIP3C3\_FinancialDocument$ XML Schema complex type (see figure 2), the following DAML+OIL is automatically generated[1]:

$$\begin{aligned}PIP3C3\_FinancialDocument \sqsubseteq\ & FinancialDocument\ \sqcap \\ & \forall\, lineItems.PIP3C3\_LineItem \\ & \geq 1\ lineItems\end{aligned}$$

[1] In the remainder of this document, we will use the Description Logic notation instead of the DAML+OIL syntax as it allows for more concise expressions.

```
<xs:complexType name="PIP3C3_FinancialDocument" abstract="true">
  <xs:annotation>
    <xs:documentation>An abstract used to capture commonalities between Invoices and InvoiceMemos
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="financialdoc:FinancialDocument">
      <xs:sequence>
        <xs:element name="lineItems" type="PIP3C3_LineItem" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Figure 2: An example of XML Schema PIP definition

1. The root schema element, complexType definitions, model group definitions and attributeGroup definitions are mapped to DAML+OIL classes.
2. Named 'simpleType' definitions stay untouched; anonymous simpleType definitions are assigned a unique name and copied to a separate datatype file.
3. Complex type elements are mapped to DAML+OIL object properties; simple type elements and attributes are mapped to DAML+OIL and attributes are mapped to DAML+OIL datatype properties.
4. Type and occurrence specifiers of elements and attributes are mapped to an intersection of DAML+OIL property type and cardinality restrictions.
5. Extensions and restrictions are mapped to a DAML+OIL subClass relationship.
6. Groups with a 'choice' compositor are mapped to a DAML+OIL disjointUnionOf collection.
7. Groups with an 'all' or 'sequence' compositor are mapped to a DAML+OIL intersectionOf collection.
8. SubstitutionGroup relationships are mapped to a DAML+OIL subProperty relationship
9. Names of components are always mapped to an URI composed of the schema targetNamespace, '#' and the component's name.

Figure 1: XML Schema to DAML+OIL Mapping

It is now possible to express constraints of a semantic nature on business objects such as $PIP3C3\_FinancialDocument$.

3.2 The Constraint Knowledge Base

We have identified in section 2 that a constraint on a business object depends on its deployment context. A deployment context is characterised by: the particular PIP document the business object it appears in, the buyer and seller trading partners and the business process used.

We define a simple ontology which models the deployment contexts in which a constraint can apply. We create 3 DAML+OIL classes, $Document$, $Partner$ and $Process$, and 4 properties $document$, $buyer$, $seller$ and $process$. The deployment context ontology is populated with subclasses and instances of the 3 classes mentionned above. For example, $PIP3C3$ is an instance of $Document$, $EuropeanPartner$ is a subclass of $Partner$.

$$Context \doteq \forall document.Document \sqcap \forall buyer.Partner \sqcap \forall seller.Partner \sqcap \forall process.Process$$

A deployment context is created by subclassing the $Context$ class and adding restrictions on one or more of the four properties, allowing the specification of restricted contexts. For example, the $buyer$ property can be restricted to have the sub-class $EuropeanPartner$, to allow the definition of a constraint which applies to all European Buyers.

To represent constraints, we also create the $Constraint$ class and the $inContext$ property.

$$Constraint \doteq \forall inContext.Context$$

A constraint on class C in context Ctx is defined as follow:

$$C \sqcap \forall inContext.Ctx \sqsubseteq \ldots$$

As an example, the constraint that restricts the class $PIP3C3\_FinancialD$ in context Ctx to have at most 10 $lineItems$ elements and at least 1 $soldTo$ elements could be written:

$$PIP3C3\_FinancialDocument \sqcap \forall inContext.Ctx \sqsubseteq \; \leq 10\, lineItems \sqcap \exists\, soldTo$$

We now give a more complex example constraining the class $PIP3C3\_FinancialDocument$ in context Ctx: if the $PIP3C3\_Financia$ instance has any $lineItems$ elements, these $lineItems$ elements must have at least one $totalLineItemAmount$ element. This is expressed as follow:

$$PIP3C3\_FinancialDocument \sqcap \forall inContext.Ctx \sqsubseteq \forall lineItems.(\exists\, totalLineItemAmount)$$

As an experiment RosettaNet has been using OCL to represent semantic constraints on PIP document specifications. Our study of these specifications show that these constraints only use a subset of OCL and can all be represented in DAML+OIL. We have successfully used OilEd [1] to model these constraints in DAML+OIL.

DAML+OIL is a very powerful ontology language but also is quite complex for non-expert users. One of the goal of the Nile Constraint Editor is to hide this complexity from the RosettaNet implementors. An analysis of RosettaNet deployments has been carried out to determine the kinds of constraint commonly applied to documents by businesses, and 3 key classes have been identified. Here we present templates for each class, together with an example constraint of that class.

1. Cardinality constraints: because of the diversity of deployment scenarios, the RosettaNet specification may leave a lot of flexibility for the cardinality of some fields (for instance 0..∞). However, specific deployment contexts may impose more constrained cardinalities.

   To restrict the maximum cardinality of *lineItems* to 10 on all *Invoice* classes in context *Ctx*, the tool generates the following statement:

$$Invoice \sqcap \forall\, inContext.Ctx \quad \sqsubseteq \quad \leq 10\, lineItems$$

2. Data format constraints: for the same reasons, the format of some data needs to be constrained. Common examples include the size of a string or the format of a date. In DAML+OIL terms, a datatype property needs to have a more restricted XML Schema type. The tool generates a restricted XML Schema type and constraints the datatype property to this newly defined type.

3. Interdependency of fields: the presence of a field, or the value of a field, may imply a cardinality constraint of a data format constraint.

3.3 Validation and inferencing on XML document

At design time, one can use the XML Schema to DAML+OIL mapping tool to create a skeleton ontology. One can also extend or refine this generated ontology. At runtime, when XML documents are being processed, one would like to perform validation and/or inferencing on these instance documents based on the extended ontology.

The idea is to have a 2-phase validation process:

- validation of the syntax of an incoming document according to an XML Schema with a validating XML parser;
- validation of the semantic constraints according with a Description Logic reasoner.

In order to validate the syntax, we need to generate an XML Schema that conveys all the syntactic restrictions, i.e. the data format constraints, that have been added in the knowledge base. This new XML Schema is almost identical to the original one that was inputed in the system: the only difference is that some simple types are now more restrictive since they have been constrained by various facets (for instance, length, regular expression or maximum value).

DAML+OIL reasoners are meant to do inference on RDF data. Since XML and RDF have different modeling foundation, we have developed a tool that translates XML documents to RDF data so that they can be processed by a DAML+OIL reasoner. This tool outputs an XML tree model to an RDF directed graph representation, and since a tree is a specialization of a graph this is always possible. Also the tool assumes that the XML documents validate

1. Create an RDF resource representing the XML document.
2. Add an `rdf:type` property to the RDF resource. Its value is the URI of the DAML+OIL class corresponding to the XML Schema type of the XML element.
3. Each sub-elements and attributes are translated to RDF properties on this RDF resource.
4. If an element is a leaf node (i.e. it contains data), the data is represented as an RDF literal on this property. Otherwise if an element contains attributes and/or sub-elements, it is transformed to an anonymous resource which becomes the value of the respective property and which is recursively transformed starting at step 2.

Figure 3: XML to RDF Mapping

an XML Schema. This is an important consideration as it allows a generic mapping. We give an overview of the mapping in figure 3 but a detailed explanation of the mapping is out of scope of this paper.

3.4 Implementation Details

4. USING NILE

The NILE tool can be used to commission a new B2B partnership, and to manage an existing one. Here we present the stages this process goes through, and show how NILE is used at each stage.

1. For a new relationship to be set up between two partners, they must first identify the appropriate PIPs and associated documents which will be transferred. Given this (assuming the PIP is next-generation), they will have access to the RosettaNet XML Schema templates for the documents. The XML Schema are loaded into the Nile Constraint Editor and automatically translated into a DAML+OIL knowledge base. For instance to set up a relationship involving PIP3C3, the PIP3C3 XML Schema (and the other 4 imported XML Schemas) need to be loaded in the Nile Constraint Editor. If a business has already used this PIP with another partner, appropriate information will already appear in their knowledge base, so they can skip this stage.
2. The partners augment the set of constraints with personal constraints which are imposed by their internal business processes and the specifics of the relationship they are trying to set up. These should represent constraints which would require business re-engineering to alter, not simply preferences. Legacy systems often impose hard constraints that cannot be altered. Previously entered constraints that are compatible with the current deployment context are automatically inherited.
3. When both partners have prepared their constraints, they can determine if their processes are potentially compatible. They do this by sharing constraints, and determining if the union of both constraint sets are satisfiable. A DL reasoner, such as Fact or Racer, is used to verify this. This can either be done by a third party, or by one or both of the partners. If the constraints are not satisfiable, it means there is a fundamental mismatch between the two business processes, and

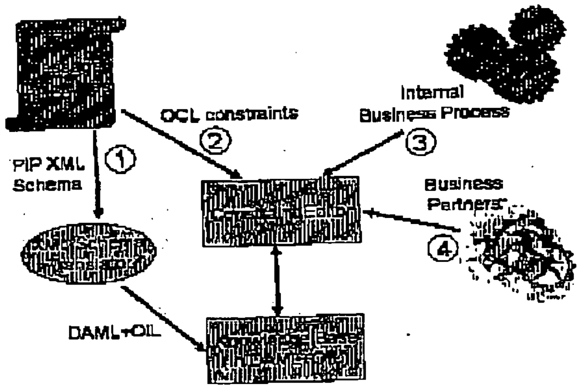

Figure 4: The Nile Constraint Editor

re-engineering will be required to achieve compatibility. The two partners will need to enter into off-line negotiations to determine how to handle this. When one or both partners have altered their business process, they should adjust their constraints to reflect this, and return to stage 1. If the constraints are satisfiable, the intersection of the two defines a subclass of the document definition which is acceptable to both parties.[**EG**] This is used as input to the next stage.

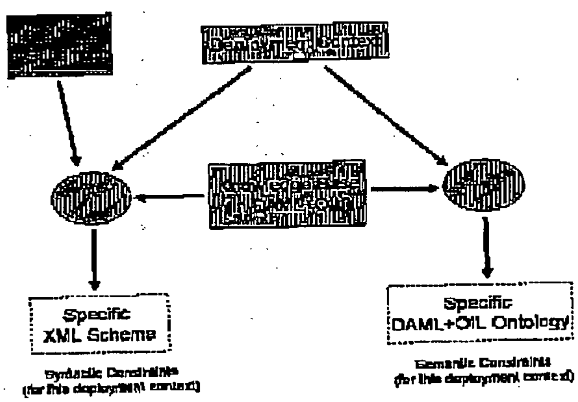

Figure 5: Deploying a PIP in a given context

4. Each partner now needs to assimilate the other partner's constraints into their constraint knowledge base. Obviously, they do not want to apply these constraints every time they send a document of this type, but only in the appropriate context. The least general way of doing this is to apply a new constraint only when sending a specific document to this specific partner, in the context of this PIP. This approach would work, and generation of the appropriate constraints could be straightforwardly automated. However, doing this loses some knowledge which is captured during the constraint generation process. Often, a constraint applies to a feature of the business process which may appear in many documents and many PIPs. To encourage maintainability and re-usability, it is best to generate a single constraint which applies in a more general context. [**EG**] The context may even go beyond the specific trading partner, if there are a certain class of partners which all need similar constraints. [**EG**] Currently, this generalisation of constraints is carried out by the developers.

5. For a given relationship and PIP, Nile is used to generate templates for all documents which will be exchanged. Initially, these templates are DAML/OWL classes generated by applying the constraint knowledge base in the specific context of this relationship. [**EG**] These templates are converted on a 'best effort' basis via the translation suite into XML Schema. [**EG**] Any constraints in DAML/OWL which can't be represented in the Schema format will be output as comments within the schema.[**EG**]

6. Developers use Contivo, together with the XML Schema, to enable generation of documents at runtime as required by the execution of a PIP.

7. Optionally, runtime validation can be carried out. When an XML document is generated by the Contivo tool, NILE can check that it does indeed satisfy the constraints in the Knowledge Base. It does this by using the translation tool to convert the XML document into an RDF representation, and using a DAML reasoner such as FACT or RACER to check that the RDF structure is indeed an instance of the required DAML class. [**EG???]

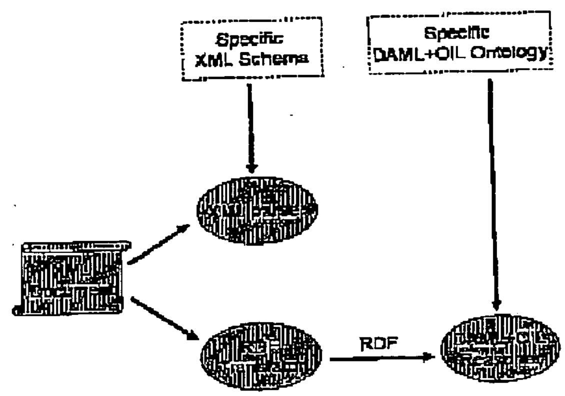

Figure 6: The Nile System at runtime

In this way, developers can use NILE in conjunction with existing products to rapidly set up new RosettaNet relationships.

5. RELATED WORK

6. CONCLUSIONS AND FUTURE WORK

The NILE tool allows developers to explicitly represent the constraints on interactions in different contexts and to re-use constraints between messages and businesses. This makes the process of setting up new relationships faster and the resulting software is more reliable and re-usable.

The approach we have taken is evolutionary, in that it accepts RosettaNet in its current form. However, we believe that our experiences can provide valuable input into future enhancements of RosettaNet. In particular, we believe that it should adopt DAML/OWL as the constraint language to formally specify PIP messages. The increased expressivity will avoid unnecessary ambiguity in the specification. Furthermore, it will allow new tools to be developed which support business interaction by reasoning with the constraints.

We have developed a suite of tools to allow the use of DAML+OIL rich semantic modeling features in XML documents and their associated XML Schemas. While these tools have their origin in the Nile system, we believe them to be generic and think they could be applied to other situations.

7. ACKNOWLEDGMENTS

Thanks to Matthias Ferdinand for his contributions on the XML Schema to DAML+OIL tool set. Thanks to Abdel Boulmakoul, Jeremy Carroll, Michel Morciniec and Giacomo Piccinelli.

8. REFERENCES

[1] S. Bechhofer, I. Horrocks, C. Goble, and R. Stevens. OilEd: a reason-able ontology editor for the semantic web. In *Working Notes of the 2001 Int. Description Logics Workshop (DL-2001)*, pages 1–9, 2001.

[2] P. V. Biron and A. Malhotra. XML Schema Part 2: Datatypes. W3C Recommendation, May 2001.

[3] T. Bray, J. Paoli, and C. M. Sperberg-McQueen. Extensible Markup Language (XML) 1.0 (second edition). W3C Recommendation, October 2000.

[4] D. Brickley and R. Guha. Resource Description Framework (RDF) Schema Specification 1.0. W3C Candidate Recommendation, March 2000.

[5] Contivo. http://www.contivo.com.

[6] B. Hofreiter, C. Huemer, and W. Winiwarter. Towards syntax-independent B2B. *ERCIM News*, 51:25–26, October 2002.

[7] I. Horrocks. Reasoning with expressive description logics: Theory and practice. In A. Voronkov, editor, *Proc. of the 18th Int. Conf. on Automated Deduction (CADE-18)*, number 2392 in Lecture Notes in Artificial Intelligence, pages 1–15. Springer-Verlag, 2002.

[8] J. Hunter and C. Lagoze. Combining RDF and XML Schemas to Enhance Interoperability Between Metadata Application Profiles. In *Proceedings 10th International World Wide Web Conference (WWW10)*, 2001.

[9] Joint US/EU ad hoc Agent Markup Language Committee. DAML+OIL (March 2001), http://www.daml.org, 2001.

[10] O. Lassila and R. Swick. Resource Description Framework (RDF) Model and Syntax Specification. W3C Recommendation. W3C Recommendation, February 1999.

[11] M. Mani, D. Lee, and R. Muntz. Semantic Data Modeling using XML Schemas. In *Proceedings 20th Intl. Conf. on Conceptual Modeling (ER)*, 2001.

[12] Object Management Group. *Object Constraint Language Specification*, September 2001. Published as part of [13].

[13] Object Management Group. *Unified Modeling Language Specification, Version 1.4*, September 2001.

[14] P. Patel-Schneider and J. Simon. The Yin/Yang Web: XML Syntax and RDF Semantics. In *Proceedings 11th International World Wide Web Conference (WWW2002)*, 2002.

[15] RosettaNet. http://www.rosettanet.org.

[16] RosettaNet. *PIP 3C3: Notify of Invoice*, November 2001.

[17] H. S. Thompson, D. Beech, M. Maloney, and N. Mendelsohn. XML Schema Part 1: Structures. W3C Recommendation, May 2001.